

Real-time Threat Feeds user guide

Generated on March 02, 2026

Contents

1	Introduction	3
1.1	A summary of common features	3
1.2	The feed descriptions and URLs	3
1.2.1	Domain discovery	3
1.2.2	Domain hotlist	3
1.2.3	Domain risk	4
1.2.4	Newly active domains (NAD)	4
1.2.5	Newly observed domains (NOD)	4
1.2.6	Newly observed hostnames (NOH)	4
1.2.7	Parsed domain RDAP (Registration Data Access Protocol)	5
1.3	A summary of access methods	5
2	Getting started with Threat Feeds	5
2.1	The authentication quick start	5
2.2	The Feed API quick start	5
2.3	The Download API quick start	6
3	API requirements	7
4	API usage limits	7
5	Recommended connection speeds	7
6	Authentication methods	7
6.1	API key (header) authentication	7
6.1.1	Required API key authentication parameters	7
6.1.2	Example request with API key authentication	8
6.2	HMAC signature authentication	8
6.2.1	Constructing the HMAC signature	8
6.3	API key and secret authentication	9
6.3.1	Required parameters for API key and secret	9
6.3.2	Example request with API key and secret	9
6.4	Response policy zone (RPZ) authentication	9

7	The Feed API	9
7.1	The Feed API base URL	9
7.2	The Feed API endpoints	9
7.3	The Feed API header options	10
7.4	The Feed API common GET request parameters	10
7.4.1	afterparameter	10
7.4.2	beforeparameter	10
7.4.3	domainparameter	10
7.4.4	fromBeginningparameter	11
7.4.5	headersparameter	11
7.4.6	malware_minparameter	11
7.4.7	overall_minparameter	12
7.4.8	phishing_minparameter	12
7.4.9	proximity_minparameter	12
7.4.10	sessionIDparameter	12
7.4.11	spam_minparameter	13
7.4.12	topparameter	13
7.5	The Feed API common DELETE request parameters	13
7.5.1	sessionIDparameter	13
7.6	The Feed API query examples	13
7.7	The Feed API response codes	14
7.8	Feed API response structure and examples	14
7.8.1	NAD, NOD, NOH, and domain discovery response structure	14
7.8.2	Domain hotlist response structure	14
7.8.3	Domain risk response structure	16
7.8.4	Domain RDAP response structure	17
8	The Download API	19
8.1	The Download API base URL	19
8.2	The Download API endpoints	20
8.3	The Download API common GET query parameters	20
8.3.1	api_keyparameter	20
8.3.2	api_usernameparameter	20
8.3.3	app_nameparameter	20
8.3.4	app_partnerparameter	20
8.3.5	app_versionparameter	20
8.3.6	limitparameter	21
8.3.7	signatureparameter	21
8.3.8	timestampparameter	21
8.4	The Download API query examples	21
8.5	The Download API response codes	21
8.6	The Download API common response parameters	21
8.7	The Download API response structure and example	22
8.7.1	API response	22
8.7.2	File contents	23
8.8	The S3 delivery	23
9	Response Policy Zone (RPZ)	23
10	Python SDK	23

Note

View the updated [Threat Feeds documentation](#).

1 Introduction

Real-Time Threat Feeds provide data on the different stages of the domain lifecycle: from first-observed in the wild to newly re-activated after a period of quiet. You can access current feed data in real-time or retrieve historical feed data through separate APIs (Application Programming Interfaces). Some feeds also offer data for DNS (Domain Name System) firewalls in Response Policy Zone (RPZ) format.

1.1 A summary of common features

- **Stable, static URL endpoint:** fetch the latest feed data with the same query parameters.
- **Configurable polling frequency:** fetch as often as you like, up to every 60 seconds.
- **5-day data retention time for Feed API:** current feed data is easily accessible.
- **Access historical data via Download API:** never lose out on data missed accidentally.
- **Session management:** Pick up where you left off, without missing or duplicating events.
- **Domain pattern filtering:** Use the `domain` query parameter to filter a feed based on domain name patterns that are important to you, without extra downstream data processing.

1.2 The feed descriptions and URLs

1.2.1 Domain discovery

New domains as they're discovered in domain registration information, observed by our global passive DNS sensor network, or reported by trusted third parties.

Feed endpoint: `api.domaintools.com/v1/feed/domaindiscovery/`

Download endpoint: `api.domaintools.com/v1/download/domaindiscovery/`

Jump to: [Authentication](#) | [Feed API](#) | [Download API](#)

1.2.2 Domain hotlist

Domains with high [Domain Risk Scores](#) that have also been active within 24 hours.

Each entry in the Domain Hotlist is an apex-level domain. The feed emits domains when both of these conditions are met within a 24-hour period:

- Observed by passive DNS
- Assigned a [Domain Risk Score](#) of ≥ 70 Proximity OR ≥ 90 Phish OR ≥ 90 Malware OR ≥ 90 Spam

Domain Risk Scores have an expiration time of 24 hours:

- When a domain receives a high Domain Risk Score and activity was observed in passive DNS over the previous 24 hours, the feed emits the domain immediately. The expiration time is 24 hours after the first of those two events.
- If a domain receives a high Domain Risk Score but has no passive DNS activity over the past 24 hours, the system waits for the Domain Risk Score's 24-hour period to complete. If passive DNS activity is observed on a domain that already has a high and live Domain Risk Score, the feed emits the domain. The expiration time is 24 hours from the original risk observation.
- The feed doesn't emit risky domains without passive DNS activity.

The Domain Hotlist includes an [expires response field](#) with the expiration time in UTC (Coordinated Universal Time). We consider the score invalid after 24 hours. We recommend that you delete expired scores or exclude them from detections or queries.

Feed endpoint: `api.domaintools.com/v1/feed/domainhotlist/`

Download endpoint: `api.domaintools.com/v1/download/domainhotlist/`

Jump to: [Authentication](#) | [Feed API](#) | [Download API](#)

1.2.3 Domain risk

Real-time updates to [Domain Risk Scores](#) for apex domains, regardless of observed traffic.

The feed emits domains at the moment they reach a combined [Domain Risk Score](#) of 70+.

We recommend that you stop using Domain Risk entries older than 24 hours (check the [timestamp value](#)). The domainrisk feed doesn't have the expires field.

Feed endpoint: `api.domaintools.com/v1/feed/domainrisk/`

Download endpoint: `api.domaintools.com/v1/download/domainrisk/`

Jump to: [Authentication](#) | [Feed API](#) | [Download API](#)

1.2.4 Newly active domains (NAD)

Domains observed in passive DNS to be newly active in the latest lifecycle of the domain. This includes domains observed either for the first time or after an inactive period of at least 10 days.

The feed emits apex-level domains (for example, `example.com` but not `www.example.com`) as it observes them.

Feed endpoint: `api.domaintools.com/v1/feed/nad/`

Download endpoint: `api.domaintools.com/v1/download/nad/`

Jump to: [Authentication](#) | [Feed API](#) | [Download API](#)

1.2.5 Newly observed domains (NOD)

Domains observed for the first time in passive DNS.

Domains are apex-level (for example, `example.com` but not `www.example.com`), and the feed emits them as they're observed.

Feed endpoint: `api.domaintools.com/v1/feed/nod/`

Download endpoint: `api.domaintools.com/v1/download/nod/`

Jump to: [Authentication](#) | [Feed API](#) | [Download API](#)

1.2.6 Newly observed hostnames (NOH)

Hostname resolutions that we observe for the first time with our global passive DNS sensor network.

The feed emits hostnames as they're observed.

Feed endpoint: `api.domaintools.com/v1/feed/noh/`

Download endpoint: `api.domaintools.com/v1/download/noh/`

Jump to: [Authentication](#) | [Feed API](#) | [Download API](#)

1.2.7 Parsed domain RDAP (Registration Data Access Protocol)

Changes to global domain registration information, populated by the Registration Data Access Protocol (RDAP). This feed complements the 5-Minute WHOIS Feed as registries and registrars switch from WHOIS to RDAP.

Feed endpoint: `api.domaintools.com/v1/feed/domainrdap/`

Download endpoint: `api.domaintools.com/v1/download/domainrdap/`

Jump to: [Authentication](#) | [Feed API](#) | [Download API](#)

The Domain RDAP feed doesn't accept the `text/csv` Accept header.

1.3 A summary of access methods

- **Real-time Feed API:** Returns results based on absolute or relative times. See the [Feed API Quick Start](#), [authentication](#), and [Feed API](#) sections.
- **Download API:** The download API returns archives of past results as signed, temporary AWS (Amazon Web Services) S3 (Simple Storage Service) file links. See the [Download API Quick Start](#), [authentication](#), and [Download API](#) sections.
- **Response Policy Zone (RPZ):** Feed contents delivered as real-time Response Policy Zones that populate DNS firewalls. See the [Response Policy Zone](#) section. The following feeds are currently available by RPZ: Newly Active Domains, Newly Observed Domains.

2 Getting started with Threat Feeds

This section walks you through the basic steps to start using Threat Feeds. You'll learn how to authenticate your requests, fetch real-time data with the Feed API, and retrieve historical data with the Download API. Each subsection provides a quick example to get you started.

2.1 The authentication quick start

You can [authenticate](#) with either API Header or API Key and Secret methods. The following example shows API header authentication using `curl` and a sessionID called `mySIEM`. By default, the API returns one hour of results:

```
curl -H 'X-API-Key: MY_API_KEY'  
↪ 'https://api.domaintools.com/v1/feed/nod/?sessionID=mySIEM'
```

To authenticate with the API Key and Secret method, provide your `api_username` and `api_key` as query parameters. The following example uses `curl` with the same sessionID (`mySIEM`):

```
curl 'https://api.domaintools.com/v1/feed/nod/?api_key=MY_API_KEY&api_username=MY_  
↪ API_USERNAME&sessionID=mySIEM'
```

2.2 The Feed API quick start

The [Feed API](#) standard access pattern is to periodically request the most recent feed data, as often as every 60 seconds. Specify the range of data you receive in one of two ways:

- With `sessionID`: Make a call and provide a new `sessionID` parameter of your choosing. By default, the API returns the past hour of results.
 - Each subsequent call to the API using your `sessionID` returns all data since the last call.
 - Any single request returns a maximum of 10M results. Requests that exceed 10M results return an HTTP (Hypertext Transfer Protocol) 206 response code. Repeat the same request (with the same `sessionID`) to receive the next tranche of data until you receive an HTTP 200 response code.
 - You can delete this session ID to clear the saved offset by using an HTTP DELETE call.
- Or specify the time range in one of two ways:
 - Use an `after=-60` query parameter, where (in this example) `-60` indicates the previous 60 seconds.
 - Use `after` and `before` query parameters for a time range, with each parameter accepting an ISO-8601 UTC formatted timestamp (a UTC date and time of the format `YYYY-MM-DDThh:mm:ssZ`).

Optional details:

- The default response format is Newline-Delimited JSON (NDJSON), also known as JSON Lines (JSONL).
 - To enable CSV (Comma-Separated Values) format responses, set the `Accept` header to `text/csv`. You can also add `&headers=1` to the query parameters to include column headers as the first line in the response. The Parsed Domain RDAP Feed doesn't support `text/csv`.
- To limit the response payload for testing, add a `top=N` query parameter, where `N` is the number of results to return.
- To add server-side domain filtering, use the `domain` query parameter to return a specific domain or domains where a specific substring is present.
 - Exact match example: `domain=example.com`
 - Partial match example: `domain=*example*`
 - Sometimes you'll need to URL-encode the asterisk to `%2A` depending on the method used to query. In a `curl` command, you can use an asterisk (`*`) if you single-quote the URL.

2.3 The Download API quick start

The Download API returns short-lived, signed URLs for pairs of files containing historical feed data that change each hour. The `{feed_short_name}` is one of: `nod`, `nad`, `noh`, `domaindiscovery`, `domainhotlist`, `domainrisk`, or `domainrdap`:

- A **data** file: `{feed_short_name}/{YYYY-MM-DD}/{feed_short_name}-{YYYYMMDD}. {starthour:HH00}-{endhour:HH00}.json.gz`
- A **checksum** file: `{feed_short_name}/{YYYY-MM-DD}/{feed_short_name}-{YYYYMMDD}. {starthour:HH00}-{endhour:HH00}.json.gz.sha256`

Here is an example `curl` command to retrieve the list of NOD files available to download:

```
curl 'https://api.domaintools.com/v1/download/nod?api_key=MY_API_KEY&api_
↳ username=MY_API_USERNAME'
```

3 API requirements

Before you can access threat feed data, you need proper credentials and licensing. This section outlines what you'll need to get started.

You'll need a license to one or more DomainTools feeds, and API credentials. Your required API credentials will vary with your authentication method, detailed below.

Obtain your API credentials from your group's API administrator. API administrators can manage their API keys at <https://research.domaintools.com>, selecting the drop-down account menu and choosing API admin.

4 API usage limits

Real-time feeds have the following rate limits:

- 2 queries per minute
- 120 queries per hour

If you exceed these limits, the API returns an error.

5 Recommended connection speeds

For best performance, we recommend the following connection speeds:

- 10 Mbps (1.25 MB/s): For most feeds with typical use.
- 100 Mbps (12.5 MB/s) or greater: For higher-volume feeds and heavy use.

(Mbps = megabits per second) (MB/s = megabytes per second)

6 Authentication methods

You can authenticate to the Threat Feeds API using three different methods, each with different security and implementation characteristics. Choose the method that best fits your security requirements and technical environment.

Authentication is available via HTTP header, HMAC (Hash-based Message Authentication Code), or API key and secret. If you need to obtain credentials or you're unsure about your authentication options, contact enterprisesupport@domaintools.com.

6.1 API key (header) authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and authenticates your requests.

6.1.1 Required API key authentication parameters

X-Api-Key: MY_API_KEY

6.1.2 Example request with API key authentication

```
curl -H 'X-API-Key: MY_API_KEY'  
'https://api.domaintools.com/v1/feed/nod/?after=-60'
```

6.2 HMAC signature authentication

HMAC authentication is a secure alternative to API key-based methods. It requires signing each request with a SHA1 HMAC digest derived from your API secret. This provides integrity and authenticity without exposing credentials directly in the request.

This method is recommended for systems where authentication credentials should not be stored in plain text or included directly in request URLs.

api_key: API Key

api_username: API Username

signature: HMAC-SHA1 signature of api_username, timestamp, and request path

timestamp: ISO 8601 formatted UTC time (for example, 2025-06-01T15:30:00Z)

6.2.1 Constructing the HMAC signature

The HMAC signature is computed as follows:

```
signature = HMAC-SHA1(api_key, api_username + timestamp + uri_path)
```

Example Python signing function:

```
import hmac  
import hashlib  
  
def sign(api_username, api_key, timestamp, uri):  
    params = f"{api_username}{timestamp}{uri}"  
    return hmac.new(  
        api_key.encode("utf-8"), params.encode("utf-8"), hashlib.sha1  
    ).hexdigest()
```

Example Python HMAC request:

```
import os  
import datetime  
import urllib.parse  
import requests  
  
api_username = os.environ["API_USERNAME"]  
api_key = os.environ["API_KEY"]  
uri = "/v1/feed/nod/"  
host = os.environ.get("HOST", "https://api.domaintools.com/")  
timestamp =  
↳ datetime.datetime.now(datetime.timezone.utc).strftime("%Y-%m-%dT%H:%M:%SZ")  
signature = sign(api_username, api_key, timestamp, uri)  
  
response = requests.get(  
    urllib.parse.urljoin(host, uri),
```

```
params={
  "api_username": api_username,
  "signature": signature,
  "timestamp": timestamp,
  "sessionID": "mySIEM", # required parameter for queries
  "top": 1 # returns a single result, for testing
},
)
```

6.3 API key and secret authentication

This is the easiest and most insecure authentication scheme to implement: each request contains the full API key and API secret. We typically recommend using [API Header](#) authentication instead. If you're unsure about your authentication options, contact enterprisesupport@domaintools.com.

6.3.1 Required parameters for API key and secret

api_username: API username

api_key: API key

6.3.2 Example request with API key and secret

```
https://api.domaintools.com/v1/feed/nod/?after=-60&api_username=&api_key=xxxxxx
```

6.4 Response policy zone (RPZ) authentication

See the [Response Policy Zone](#) for more information.

7 The Feed API

The Feed API provides real-time access to current threat feed data. Use this API to poll for the latest feed updates at regular intervals, maintain a session to track your position in the feed, and filter results based on your specific needs.

7.1 The Feed API base URL

```
api.domaintools.com/v1/feed/
```

7.2 The Feed API endpoints

- **Domain Discovery:** `domaindiscovery`
- **Domain Hotlist:** `domainhotlist`

- **Domain RDAP:** domainrdap
- **Domain Risk:** domainrisk
- **Newly Active Domains:** nad
- **Newly Observed Domains:** nod
- **Newly Observed Hostnames:** noh

7.3 The Feed API header options

If you omit the Accept header, the API returns results in JSON Lines format (application/x-ndjson).

Clients who wish to specify the response format explicitly can use the Accept header:

Accept: `application/x-ndjson`: Get results in JSON Lines format.

Accept: `text/csv`: Get results in CSV format. Not available with the Download API, or in the domainrdap Feed API endpoint. Optionally adding `&headers=1` to the query parameters includes column headers as the first line in the response.

Note that the Domain RDAP feed doesn't accept the `text/csv` Accept header.

7.4 The Feed API common GET request parameters

7.4.1 after parameter

Type: Integer or string

Valid values: - Integer: -1 to -432,000 - String: ISO 8601 datetime string in UTC (Coordinated Universal Time) form

Description: The start of the query window, inclusive. When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp.

Example: `after=-60` or `after=2024-10-16T10:20:00Z`

Required: Yes, if `before` or `sessionID` not provided

Available for: All feeds

7.4.2 before parameter

Type: Integer or string

Valid values: - Integer: -1 to -432,000 - String: ISO 8601 datetime string in UTC form

Description: The end of the query window, inclusive. When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp.

Example: `before=-120` or `before=2024-10-16T10:20:00Z`

Required: Yes, if `after` or `sessionID` not provided

Available for: All feeds

7.4.3 domain parameter

Type: String

Valid values: Domain character set restricted by the DNS (Domain Name System) specification (Letters, Digits, Hyphens). International characters should be specified in punycode. A trailing dot is acceptable. You can include multiple domain filters in a request.

Description: Filter for an exact domain or a domain substring by prefixing or suffixing your string with *. Multiple parameters are supported. The URL-encoded version of * (%2A) may be required.

Example: domain=*apple*&domain=*microsoft*

Required: No

Available for: All feeds

7.4.4 fromBeginning parameter

Type: Boolean

Valid values: true, false

Description: Functions with new Session IDs to return the first hour (rather than the last). Returns an error if Session ID already exists.

Example: fromBeginning=true

Required: No

Available for: All feeds

7.4.5 headers parameter

Type: Boolean

Valid values: 0, 1

Description: Adds a header to the first line of response when text/csv is set in header parameters.

Example: headers=1

Required: No

Available for: All feeds

7.4.6 malware_min parameter

Type: Integer

Valid values: 1-99

Description: Filter domains for malware [domain risk scores](#) greater than or equal to this value.

Example: malware_min=75

Required: No

Available for: Domain Hotlist, Domain Risk

7.4.7 overall_min parameter

Type: Integer

Valid values: 1-99

Description: Filter domains for overall [domain risk scores](#) greater than or equal to this value.

Example: overall_min=75

Required: No

Available for: Domain Hotlist, Domain Risk

7.4.8 phishing_min parameter

Type: Integer

Valid values: 1-99

Description: Filter domains for phishing [domain risk scores](#) greater than or equal to this value.

Example: phishing_min=75

Required: No

Available for: Domain Hotlist, Domain Risk

7.4.9 proximity_min parameter

Type: Integer

Valid values: 1-99

Description: Filter domains for proximity [domain risk scores](#) greater than or equal to this value.

Example: proximity_min=75

Required: No

Available for: Domain Hotlist, Domain Risk

7.4.10 sessionID parameter

Type: String

Valid values: 1-64 alphanumeric characters ([a-zA-Z0-9-]+)

Description: A string that serves as a unique identifier for the session, used for resuming data retrieval from the last point.

Example: sessionID=mySIEM

Required: Yes, to continue where you left off

Available for: All feeds

7.4.11 spam_min parameter

Type: Integer

Valid values: 1-99

Description: Filter domains for spam [domain risk scores](#) greater than or equal to this value.

Example: spam_min=75

Required: No

Available for: Domain Hotlist, Domain Risk

7.4.12 top parameter

Type: Integer

Valid values: Positive integer, 1-1,000,000,000

Description: Limits the number of results in the response payload. When you apply this parameter to domainrisk, results are sorted by overall_risk (descending).

Example: top=10

Required: No

Available for: All feeds

7.5 The Feed API common DELETE request parameters

7.5.1 sessionID parameter

Type: String

Valid values: 1-64 alphanumeric characters ([a-zA-Z0-9-]+)

Description: A string that serves as a unique identifier for the session, used for resuming data retrieval from the last point.

Example: sessionID=mySIEM

Required: No

7.6 The Feed API query examples

Use curl and header authentication to retrieve the last 60 seconds of Newly Observed Domains:

```
curl -H 'X-Api-Key: MY_API_KEY'  
'https://api.domaintools.com/v1/feed/nod/?after=-60'
```

Use curl and header authentication to retrieve the last 4 hours of Newly Observed Domains that contain the keyword bank:

```
curl -H 'X-Api-Key: MY_API_KEY'  
'https://api.domaintools.com/v1/feed/nod/?after=-14400&domain=*bank*'
```

7.7 The Feed API response codes

200: OK: The request was successful.

206: Partial content: The request was successful, but only a portion of the data was returned.

400: Malformed request

401: Unauthorized

403: Forbidden

404: sessionID does not exist

406: Not Acceptable: The specified Accept header value is not supported. Only application/x-ndjson and text/csv are accepted.

422: Invalid header query parameter

If an API call returns a HTTP 206 response, continue submitting the same request (with the same sessionID) until the API returns a HTTP 200, signalling that all the data for the request has been delivered.

7.8 Feed API response structure and examples

7.8.1 NAD, NOD, NOH, and domain discovery response structure

NAD, NOD, NOH, and Domain Discovery return one hour of results by default.

The API returns responses in JSON-lines (JSONL), with each response containing one domain entry per line. Each entry contains a timestamp in ISO 8601 UTC form, and the domain.

domain (string): Apex-level domain for NAD, NOD, and Domain Discovery. The NOH feed returns full hostnames (for example, including subdomains). Domain character set restricted by the DNS specification (Letters, Digits, Hyphens). Example: "domain": "example.com"

timestamp (string): Discovery timestamp in ISO 8601 datetime string in UTC form. Example: "timestamp": "2024-11-15T16:14:39Z"

Example response:

```
{"timestamp": "2024-11-15T16:14:39Z", "domain": "domiantools.com"}
{"timestamp": "2024-11-15T16:14:38Z", "domain": "domsintools.com"}
{"timestamp": "2024-11-15T16:14:36Z", "domain": "edomaintools.com"}
{"timestamp": "2024-11-15T16:14:35Z", "domain": "omaintools.com"}
{"timestamp": "2024-11-15T16:14:35Z", "domain": "v-domaintools.com"}
```

7.8.2 Domain hotlist response structure

The Domain Hotlist returns 1 hour of results by default. The API emits null values.

The API returns responses in JSON-lines (JSONL), with each response containing one domain entry per line. In addition to the domain and timestamp parameters, the Domain Hotlist and Domain Risk Feeds include risk scoring:

7.8.2.1 timestamp field

Type: String (ISO 8601 UTC timestamp)

Description: The date and time of discovery in UTC

Example: "timestamp"="2024-11-15T16:14:39Z"

7.8.2.2 domain field

Type: String

Valid values: Domain character set restricted by the DNS specification (Letters, Digits, Hyphens)

Description: Apex-level domain for NAD, NOD, and Domain Discovery. The NOH feed returns full hostnames (for example, including subdomains).

Example: "domain"="example.com"

7.8.2.3 phishing_risk field

Type: Integer

Valid values: 0-100 or null

Description: Phishing [domain risk score](#)

Example: phishing_risk=80

7.8.2.4 malware_risk field

Type: Integer

Valid values: 0-100 or null

Description: Malware [domain risk score](#)

Example: malware_risk=80

7.8.2.5 spam_risk field

Type: Integer

Valid values: 0-100 or null

Description: Spam [domain risk score](#)

Example: spam_risk=80

7.8.2.6 proximity_risk field

Type: Integer

Valid values: 0-100

Description: Proximity [domain risk score](#)

Example: proximity_risk=80

7.8.2.7 overall_risk field

Type: Integer

Valid values: 0-100 or null

Description: Overall [domain risk score](#)

Example: overall_risk=80

7.8.2.8 expires field

Type: String (ISO 8601 UTC timestamp)

Description: The expiration of the entry. The expiration is 24 hours after the first of the two required events (risk or activity) is detected.

Example: "expires":"2025-08-19T19:08:58Z"

Example response via

api.domaintools.com/v1/feed/domainhotlist?sessionID=mySOC&top=3 (with header authorization):

```
{"timestamp":"2025-08-18T19:16:47Z","domain":"domiantools.com","phishing_
↪ risk":99,"malware_risk":99,"spam_risk":99,"proximity_risk":99,"overall_
↪ risk":99,"expires":"2025-08-19T19:08:58Z"}
{"timestamp":"2025-08-18T19:27:56Z","domain":"domsintools.com","phishing_
↪ risk":99,"malware_risk":99,"spam_risk":99,"proximity_risk":99,"overall_
↪ risk":99,"expires":"2025-08-19T19:23:30Z"}
{"timestamp":"2025-08-18T18:52:12Z","domain":"edomaintools.com","phishing_
↪ risk":99,"malware_risk":99,"spam_risk":99,"proximity_risk":99,"overall_
↪ risk":99,"expires":"2025-08-19T18:44:51Z"}
```

7.8.3 Domain risk response structure

The Domain Risk Feed returns one hour of results by default. The API emits null values.

The API returns responses in JSON-lines (JSONL), with each response containing one domain entry per line. In addition to the domain and timestamp parameters, the Domain Hotlist and Domain Risk Feeds include risk scoring:

7.8.3.1 timestamp field

Type: String (ISO 8601 UTC timestamp)

Description: The date and time of discovery in UTC

Example: "timestamp"="2024-11-15T16:14:39Z"

7.8.3.2 domain field

Type: String

Valid values: Domain character set restricted by the DNS specification (Letters, Digits, Hyphens)

Description: Apex-level domain for NAD, NOD, and Domain Discovery. The NOH feed returns full hostnames (for example, including subdomains).

Example: "domain"="example.com"

7.8.3.3 phishing_risk field

Type: Integer

Valid values: 0-100 or null

Description: Phishing [domain risk score](#)

Example: phishing_risk=80

7.8.3.4 malware_risk field

Type: Integer

Valid values: 0-100 or null

Description: Malware [domain risk score](#)

Example: malware_risk=80

7.8.3.5 spam_risk field

Type: Integer

Valid values: 0-100 or null

Description: Spam [domain risk score](#)

Example: spam_risk=80

7.8.3.6 proximity_risk field

Type: Integer

Valid values: 0-100

Description: Proximity [domain risk score](#)

Example: proximity_risk=80

7.8.3.7 overall_risk field

Type: Integer

Valid values: 0-100 or null

Description: Overall [domain risk score](#)

Example: overall_risk=80

Example response via api.domaintools.com/v1/feed/domainrisk?sessionID=mySOC&top=3 (with header authorization):

```
{
  "timestamp": "2025-04-22T16:08:33Z",
  "domain": "omaintools.com",
  "phishing_risk": 94,
  "malware_risk": 88,
  "spam_risk": 93,
  "proximity_risk": 80,
  "overall_risk": 94
}
{"timestamp": "2025-04-22T16:08:29Z",
  "domain": "v-domaintools.com",
  "phishing_risk": 96,
  "malware_risk": 91,
  "spam_risk": 99,
  "proximity_risk": 85,
  "overall_risk": 99
}
{"timestamp": "2025-04-22T16:08:34Z",
  "domain": "domanitools.com",
  "phishing_risk": 98,
  "malware_risk": 97,
  "spam_risk": 68,
  "proximity_risk": 72,
  "overall_risk": 98
}
```

7.8.4 Domain RDAP response structure

The Parsed Domain RDAP feed returns one hour of results by default.

The API returns responses in JSON (not NDJSON). Note that the Domain RDAP feed doesn't accept the text/csv Accept header.

Domain RDAP records for a given domain may be provided by a domain registry, registrar, or both. Domain registries maintain authoritative information about one or more top-level

domains (e.g., .com), while domain registrars manage apex domains (e.g., domaintools.com). When domain information is present from both the registry and registrar, this API presents a record containing both sets of results, as well the original raw JSON record, from both the registry and registrar.

Each response begins with either the raw *registrar* record, the raw *registry* record, or when useful information is present in both records, the response will contain both the registrar and registry record. The parsed record then follows the raw record.

timestamp (string): The date and time of discovery in UTC. ISO 8601 UTC timestamp.

Example: "timestamp"="2024-11-15T16:14:39Z"

domain (string): Apex-level domain for NAD, NOD, and Domain Discovery. The NOH feed returns full hostnames (for example, including subdomains). Domain character set restricted by the DNS specification (Letters, Digits, Hyphens). Example: "domain"="example.com"

raw_record (object): Contains the raw registry and/or registrar for the domain. See example below.

first_request_timestamp (string): The timestamp of the first request. ISO 8601 UTC timestamp. Example: "first_request_timestamp"="2025-04-23T10:30:17Z"

requests (array): Record request objects. See example below.

data (string): The raw data of the request. JSON string. Example:

```
"data"="{\"objectClassName\": \"domain\", \"handle\": \"1831890332_DOMAIN_NET-VRSN\\\"...}"
```

source_type (string): The source type of the request. Example: "source_type"="registrar"

timestamp (string): The timestamp of the request. ISO 8601 UTC timestamp. Example:

```
"timestamp"="2025-04-23T10:30:18Z"
```

url (string): The URL associated with the request. Example:

```
"url"="https://rdap.nicproxy.com/domain/domiantools.com"
```

parsed_record (object): Contains parsed information from the raw record. See nested fields below.

registrar_request_url (string): The URL for the registrar request. Example: "registrar_request_url"="https://rdap.nicproxy.com/domain/domiantools.com"

registry_request_url (string): The URL for the registry request. Example: "registry_request_url"="https://rdap.nicproxy.com/domain/domiantools.com"

In the following example using domaintools.com, both a registry and registrar record are present (note source_type). The raw records, as well as the content of the parsed record, are removed for brevity.

With the following query:

```
curl -sH 'X-API-Key: YOUR_API_KEY'  
↪ 'https://api.domaintools.com/v1/feed/domainrdap?sessionID=myS0C&top=1'
```

The following results are obtained:

```
{  
  "timestamp": "2024-11-15T00:00:19Z",  
  "domain": "domaintools.com",
```

```

"raw_record": {
  "first_request_timestamp": "2024-11-15T00:00:14Z",
  "requests": [
    {
      "data": "{RAW REGISTRY RECORD}",
      "source_type": "registry",
      "timestamp": "2024-11-15T00:00:14Z",
      "url": "https://rdap.verisign.com/com/v1/domain/domaintools.com"
    },
    {
      "data": "{RAW REGISTRAR RECORD}",
      "source_type": "registrar",
      "timestamp": "2024-11-15T00:00:16Z",
      "url": "https://enom.rdap.tucows.com/domain/DOMAINTOOLS.COM"
    }
  ]
},
"parsed_record": {
  "parsed_fields": {PARSED FIELDS},
  "registrar_request_url":
↪ "https://enom.rdap.tucows.com/domain/DOMAINTOOLS.COM",
  "registry_request_url":
↪ "https://rdap.verisign.com/com/v1/domain/domaintools.com"
}
}

```

8 The Download API

The Download API provides access to historical feed data through temporary AWS S3 file links. Use this API to retrieve archived data you may have missed or to backfill your systems with historical information. Files are organized by hour and available for 90 days.

The download API returns 90 days of historical results in the form of temporary AWS S3 files. The AWS S3 files are signed and come in pairs:

- A **data** file: {feed_short_name}/{YYYY-MM-DD}/{feed_short_name}-{YYYYMMDD}. {starthour:HH00}-{endhour:HH00}.json.gz
- A **checksum** file: {feed_short_name}/{YYYY-MM-DD}/{feed_short_name}-{YYYYMMDD}. {starthour:HH00}-{endhour:HH00}.json.gz.sha256

The API lists the files available for download, after which individual files can be downloaded from the signed URL. The hourly files are gzip-compressed JSON.

Note that setting the `limit` parameter to an odd number will deliver a data file without its checksum companion.

8.1 The Download API base URL

```
api.domaintools.com/v1/download/
```

8.2 The Download API endpoints

- **Domain Discovery:** domaindiscovery
- **Domain Hotlist:** domainhotlist
- **Domain RDAP:** domainrdap
- **Domain Risk:** domainrisk
- **Newly Active Domains:** nad
- **Newly Observed Domains:** nod
- **Newly Observed Hostnames:** noh

E.g., `api.domaintools.com/v1/download/nad/`

8.3 The Download API common GET query parameters

8.3.1 api_key parameter

Type: String

Description: API key provided by DomainTools, dashes included

Required: Yes (for key authentication)

8.3.2 api_username parameter

Type: String

Description: API username provided by DomainTools

Required: Yes (for key authentication)

8.3.3 app_name parameter

Type: String

Description: Name of your appliance, playbook, module, or combination. Useful to help with debugging.

Required: No

8.3.4 app_partner parameter

Type: String

Description: Your product name. Useful to help with debugging.

Required: No

8.3.5 app_version parameter

Type: String

Description: Your version number. Useful to help with debugging.

Required: No

8.3.6 limit parameter

Type: Integer

Description: Limit the list of signed files. Ordering of files is always descending, so the latest files are first.

Required: No

8.3.7 signature parameter

Type: String

Description: HMAC (Hash-based Message Authentication Code) hash of your request, using the MD5, SHA1, or SHA256 hashing algorithm

Required: Yes (for HMAC authentication)

8.3.8 timestamp parameter

Type: String

Description: Current timestamp for HMAC authentication, in ISO 8601 format (for example, 2025-01-10T15:44:39.118Z)

Required: Yes (for HMAC authentication)

8.4 The Download API query examples

Get the latest, single (`limit=1`) signed download URL from Newly Observed Domains using `curl`:

```
curl -sH "X-API-Key: YOUR_API_KEY"  
  -> 'https://api.domaintools.com/v1/download/nod/?limit=1'
```

8.5 The Download API response codes

200: OK: The request was successful.

400: Malformed request

401: Unauthorized

403: Forbidden

422: Invalid header query parameter

If an API call returns a HTTP 206 response, continue submitting the same request (with the same `sessionID`) until the API returns a HTTP 200, signalling that all the data for the request has been delivered.

8.6 The Download API common response parameters

download_name (string): Name of the feed.

etag (string): Entity tag (a hash of the object).

last_modified (string): Last modified date of the file in ISO 8601 format.

size (integer): Size in kilobytes (KB)

url (string): Signed AWS CloudFront download URL; see note below.

The Download API returns short-lived, signed URLs for two files containing historical feed data that change each hour, where {feed_short_name} is one of: nod, nad, noh, domainrdap, domaindiscovery:

- A data file: {feed_short_name}/{YYYY-MM-DD}/{feed_short_name}-{YYYYMMDD}.{starthour:HH00}-{endhour:HH00}.json.gz
- A checksum file: {feed_short_name}/{YYYY-MM-DD}/{feed_short_name}-{YYYYMMDD}.{starthour:HH00}-{endhour:HH00}.json.gz.sha256

Note that setting the `limit` parameter to an odd number will return a checksum file without its data file.

8.7 The Download API response structure and example

8.7.1 API response

Note that the API returns file URLs in pairs: a data file (ending in `.json.gz`) and a checksum file (ending in `.json.gz.sha256`).

```
{
  "response": {
    "download_name": "nod",
    "files": [
      {
        "name": "nod/2025-08-21/nod-20250821.1200-1300.json.gz.sha256",
        "last_modified": "2025-08-21T13:00:13+00:00",
        "etag": "\"\"ETAG\"",
        "size": "64",
        "url":
        ↪ "https://dyyl2mzrdiuqox.cloudfront.net/nod/2025-08-21/nod-20250821.1200-
        ↪ 1300.json.gz.sha256?ai=14592&ai=1708207&Expires=1755824426&Signature=SIGNATURE&Key-
        ↪ Pair-Id=KEYPAIRID"
      },
      {
        "name": "nod/2025-08-21/nod-20250821.1200-1300.json.gz",
        "last_modified": "2025-08-21T13:00:13+00:00",
        "etag": "\"\"ETAG\"",
        "size": "140725",
        "url":
        ↪ "https://d2rdimzuqylox.cloudfront.net/nod/2025-08-21/nod-20250821.1200-
        ↪ 1300.json.gz?ai=14592&ai=1708207&Expires=1755824426&Signature=SIGNATURE&Key-
        ↪ Pair-Id=KEYPAIRID"
      }
    ]
  }
}
```

8.7.2 File contents

The *.json.gz.sha256 file is a checksum containing a SHA-256 hash value used to verify the integrity of the downloaded file.

The *.json.gz file, uncompressed to a JSON file containing the same information as the Feed API.

8.8 The S3 delivery

As an alternative to the Download API, DomainTools can deliver feed data directly to your Amazon S3 bucket. This cloud-based delivery method replaces the legacy transfer server system and supports scheduled data pushes at custom intervals (such as every 5 minutes or daily).

Requirements

- An Amazon S3 bucket with write access
- S3 credentials (access key and secret key)
- Coordination with DomainTools to configure the delivery schedule and feed selection

How it works

DomainTools configures an automated sync process that transfers feed files from DomainTools' S3 infrastructure to your designated S3 bucket. The sync runs on your specified schedule, ensuring you receive feed data without polling the Download API.

Setup

To enable S3 delivery, contact enterprisesupport@domaintools.com with:

- Your S3 bucket name and region
- The feed(s) you want delivered
- Your preferred delivery schedule
- S3 credentials for write access (provided securely during setup)

DomainTools manages the sync schedule and monitors for delivery failures.

9 Response Policy Zone (RPZ)

See the [Response Policy Zone](#) documentation.

10 Python SDK

For Python integration examples and methods, consult our [Python SDK documentation](#) or the [GitHub repository](#).